Strategies for Adding Adaptive Learning Mechanisms to Rule-Based Diagnostic Expert Systems*

D.C. St. Clair**,
C. L. Sabharwal
University of MO--Rolla
Graduate Engineering Center
St. Louis, MO 63121

W.E. Bond,
Keith Hacke
McDonnell Douglas Research Laboratories
St. Louis, MO 63166

ABSTRACT

Rule-based diagnostic expert systems can be used to perform many of the diagnostic chores necessary in today's complex space systems. These expert systems typically take a set of symptoms as input and produce diagnostic advice as output. The primary objective of such expert systems is to provide accurate and comprehensive advice which can be used to help return the space system in question to nominal operation.

The development and maintenance of diagnostic expert systems is time and labor intensive since the services of both knowledge engineer(s) and domain expert(s) are required. The use of adaptive learning mechanisms to incrementally evaluate and refine rules promises to reduce both time and labor costs associated with such systems. This paper describes the basic adaptive learning mechanisms of strengthening, weakening, generalization, discrimination, and discovery. Next, basic strategies are discussed for adding these learning mechanisms to rule-based diagnostic expert systems. These strategies support the incremental evaluation and refinement of rules in the knowledge base by comparing the set of advice given by the expert system (A) with the correct diagnosis (C). Techniques are described for selecting those rules in the knowledge base which should participate in adaptive learning.

The strategies presented may be used with a wide variety of learning algorithms. Further, these strategies are applicable to a large number of rule-based diagnostic expert systems. They may be used to provide either immediate or deferred updating of the knowledge base.

INTRODUCTION

The basic architecture of rule-based diagnostic expert systems is shown in Figure 1. Symptoms describing the failure to be diagnosed are entered into the expert system via the user interface. The system then "reasons" over the set of symptoms, asking for additional information if necessary. At the conclusion of the "reasoning" process, the expert system provides suggestions for correcting the anomalies of the system under diagnosis. As can be seen in

* This work was supported by the McDonnell Douglas Independent Research and Development program and by Statement of Work WS-MDRL-2703 with the University of MO-Rolla Graduate Engineering Center.

** D. St. Clair's Bitnet address is C0567@UMRVMB.

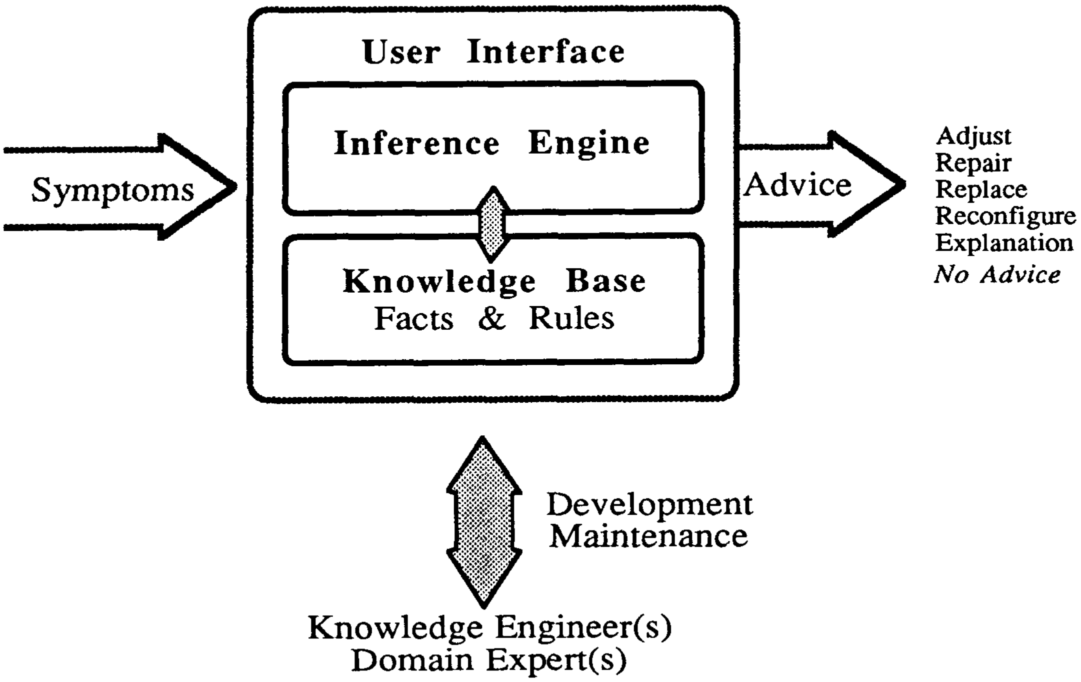

Figure 1. Rule-Based Diagnostic Expert System Architecture.

Figure 1, this advice can take one or more forms. It is also possible that the system will be unable to provide any advice. Of course, the objective of a well-designed expert system is to provide both accurate and comprehensive advice.

The basic components of a rule-based expert system are a knowledge base and an inference engine. The knowledge base consists of a set of facts and a set of rules. Although actual knowledge representations vary from application to application, the rules are logically equivalent to the form:

$$\mathbf{H} \rightarrow \mathbf{K}$$

where set **H** contains one or more elements from the description space. The description space, determined by knowledge engineers and domain experts, contains all the propositions and negated propositions used to describe the environment. Set **K** contains a set of actions to be performed. These actions may modify the knowledge base and/or cause external actions to be performed. Without loss of generality, the remaining discussion assumes that each rule conclusion contains a single action such as "replace part C6."

Clearly, expert system performance is directly related to the accuracy and comprehensiveness of rules in the knowledge base. One basic approach to evaluating rule accuracy and comprehensiveness has been to compare the system's advice with the correct diagnosis. Correct diagnoses come from domain experts themselves and/or from observing the actions required to remove anomalies from the system being repaired. Once this comparison is made, a form of learning is initiated in an effort to upgrade the knowledge base and thus improve the expert system's diagnostic accuracy. The most common approach to learning in this situation utilizes domain experts and knowledge engineers to manually revise the contents of the knowledge base. Although St. Clair et al. [10] have suggested a technique to assist in this endeavor, manual updating of rules is still time and labor intensive.

Attempts are currently under way to develop machine learning techniques which can be used to automate revision of the knowledge base. Both nonincremental and incremental techniques have been developed. Nonincremental learning techniques perform all learning at a given point in time. While the learning process may be repeated periodically, these techniques require that all accumulated test cases be processed at once [4,7]. Other techniques use incremental learning mechanisms which continuously update portions of the knowledge base [9,11]. They attempt to improve the accuracy of advice by refining the components of existing rules. Incremental learning techniques seek to improve the comprehensiveness of advice by creating new rules as necessary.

The following section outlines some of the basic adaptive learning mechanisms. It is followed by a discussion of how expert system advice is classified. Then, strategies for adding adaptive learning mechanisms to rule-based diagnostic expert systems are discussed.

BASIC ADAPTIVE LEARNING MECHANISMS

In diagnostic expert systems, adaptive learning mechanisms utilize the input symptoms, the current knowledge base, and the correct diagnosis to produce an updated knowledge base. Approaches range from simple to highly specialized. Figure 2 illustrates the basic relationship of these components. The adaptive learner continuously applies various learning mechanisms to selected rules in the knowledge base.

Adaptive learning strategies fall into one of six basic categories [1,3,9]: strengthening, weakening, unlearning, generalization, discrimination, and discovery. Strengthening and weakening mechanisms are used to reward correct rules and to penalize incorrect rules. They

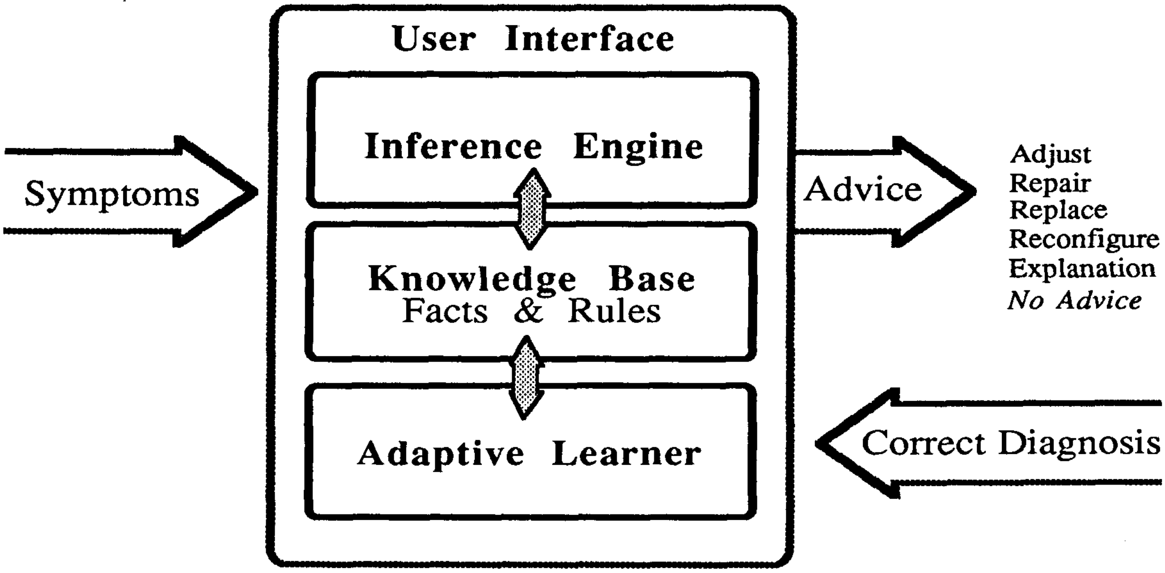

Figure 2. Adaptive Rule-Based Diagnostic Expert System Architecture.

may be implemented by keeping individual rule statistics about the number of times the rule has fired and the number of times it participated in a rule chain leading to an element of the correct diagnosis. These experience indicators can be used to effect future rule firings and to indicate the strength of the advice given in the expert system output.

Unlearning has received little attention in the literature [5]. Unlearning involves the removal of undesirable rules from the knowledge base. Experience indicators may be useful in deciding which rules to remove.

Generalization is the process of reducing the number of propositions and/or negated propositions in a rule's hypotheses. The overall result is to make the hypotheses less restrictive. The rule then becomes applicable in a larger number of situations. The need for generalization is easily identifiable in situations where a rule which should have fired did not fire. This condition, called an error of omission, usually indicates that the rule hypotheses were overly restrictive. Generalizing a set of rules may make it possible to combine several rules into one. Good generalization mechanisms are hard to define due to the difficulty of deciding which propositions can be removed from the rule hypotheses. Bundy et al. [3] provide some suggestions along this line.

Discrimination produces results which are essentially the opposite of generalization. In discrimination, propositions or negated propositions are added to a rule's hypotheses to restrict its firing. Discrimination mechanisms are usually easier to define than generalization mechanisms. Some algorithms [7,9] treat discrimination and generalization as complementary processes. Bundy et al. [3] gives an example to show that they are not fully complementary.

Discovery mechanisms utilize input symptoms and the current knowledge base to create new rules. These mechanisms are necessary whenever the expert system gives incomplete diagnostic advice or no advice at all. They must decide which propositions and negated propositions from the description space should constitute the hypotheses of each new rule. The list of symptoms is generally quite helpful in this regard. Discovery mechanisms may become quite complex in situations where extensive new rule chains must be constructed. As is indicated in a later section, the most difficult part of discovery learning is deciding when to apply it.

Both heuristic and analytical approaches are being applied in an effort to develop a good general set of adaptive learning mechanisms. The analytical approaches include the incremental and nonincremental classes of techniques mentioned earlier. Many approaches attempt to perform logical equivalents of the basic mechanisms described. In cases where knowledge relationships are not complex, a set of simple heuristics may suffice for implementing the learning mechanisms [9].

CLASSIFICATION OF EXPERT SYSTEM ADVICE

The foundation for deciding how to add adaptive learning mechanisms to rule-based diagnostic expert systems is based on comparing expert system advice with correct diagnoses. Accordingly, let **A** denote the advice set produced by a diagnostic expert system in response to a given set of input symptoms. The elements of **A** are the consequents produced as a result of one or more rule chains fired by the inference engine. As indicated earlier, assume that each rule chain terminates with a consequent containing a single piece of diagnostic advice such as "adjust

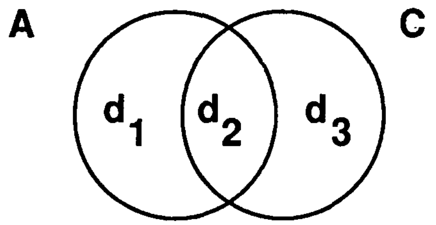

Figure 3. Comparison of Expert System Advice with Correct Diagnosis.

part D5." Further, let **C** represent the correct diagnosis. As suggested by St. Clair et al. [10], the comparison of these two sets forms the foundation for evaluating expert system output. Figure 3 illustrates the three cases which may arise.

The elements of set **A** represent components of diagnostic expert system advice while the elements of set **C** represent components of the correct diagnosis. Hence, $d_2 \in \mathbf{A} \cap \mathbf{C}$, shown in Figure 3, represents a correct piece of advice which was given by the expert system. The associated rules have produced a correct system response. The element $d_1 \in \mathbf{A} - \mathbf{C}$ represents an advice component which is incorrect. The associated rules have produced an incorrect system response and need revision. The element $d_3 \in \mathbf{C} - \mathbf{A}$ represents a case in which a component of the correct diagnosis was not included as part of the expert system's advice. This occurrence indicates a condition where the expert system has failed to provide needed advice. Since two different rule chains may produce the same advice, the components of these sets may not be distinct.

Note that the conditions illustrated by Figure 3 are a comparison of expert system output with known correct diagnosis. Such a comparison will not identify cases in which incorrect rule chains produce correct conclusions. In addition, some erroneous conditions, such as conflicting conclusions, cannot be completely uncovered by comparing the contents of sets **A** and **C**.

STRATEGIES FOR ADDING ADAPTIVE LEARNING MECHANISMS

Given a rule-based diagnostic expert system, a set of input symptoms, and the correct diagnosis for this set of symptoms, the adaptive learner must decide how and when to apply the various types of adaptive learning mechanisms described earlier (see Figure 2). These mechanisms may modify rule statistics, modify rule hypotheses, and/or create new rules. The choice of which strategies to apply is based on comparing expert system advice with the correct diagnosis as discussed above and illustrated in Figure 3.

Rule chains terminating in $d_i \in \mathbf{A} - \mathbf{C}$ represent cases in which the expert system gave incorrect advice. At least one rule in each chain has committed an error of commission by firing when it should not have fired. The strategy for adding adaptive learning mechanisms calls for discrimination to be performed to restrict the firing of the rule. In one prototype system [9], the discrimination algorithm was implemented by simply replacing the existing rule by a rule whose hypotheses were chosen from the set of input symptoms and whose conclusion was d_i. Such a simple strategy may not work well in situations in which the rule being modified participates in several other rule chains. Rules participating in more than one rule chain will have experience indicators which vary from the experience indicators of other rules in the chain.

Advice components $d_i \in \mathbf{A} \cap \mathbf{C}$ represent cases in which the expert system gave correct advice. Two situations need to be noted in this case. The first situation involves updating both experience indicators for all rules participating in the rule chain. This action does not indicate that each rule in the chain is correct but only that it has participated in a rule chain leading to a correct conclusion. The second situation calls for deciding whether or not generalization should be performed. For instance, if two or more $d_i \in \mathbf{A} \cap \mathbf{C}$ have the same conclusion, generalization may be desirable. The decision of whether or not to generalize is difficult, as the following three simple rules demonstrate.

Rule 1:
If part number = B10
measured value = 1011,
temperature range = (70 79)
intermittent = no
Then
replace part B6.

Rule 2:
If part number = B10
measured value = 1011,
Then
replace part B6.

Rule 3:
If part number = B10
temperature range = (70 79)
Then
replace part B6.

All three rules will fire whenever Rule 1 fires. The question arises as to whether generalization should be used to replace these rules by a more general rule, and if so, what should be contained in the hypotheses of the new rule. This is a difficult question at best. The experience indicators utilized in conjunction with some of the techniques mentioned in the previous section may help resolve such issues. Even though Rule 1 is the least general, if its accuracy rate is high and its times fired statistic is close to that of the other rules, it may be the case that it should be retained and the remaining rules should be removed from the knowledge base.

Those $d_i \in \mathbf{C} - \mathbf{A}$ indicate cases in which the expert system's knowledge base needs revision. If one or more rules committed an error of omission by failing to fire when they should, the knowledge base is inaccurate. On the other hand, if the knowledge base does not contain information pertinent to the diagnostic action d_i, it is incomplete. The difficulty in deciding which case applies is related directly to the complexity of the knowledge base and to the rule chains it produces.

An error of omission occurs if the knowledge base contains one or more unexecuted rule chains which terminate in the appropriate diagnostic action $d_i \in \mathbf{C} - \mathbf{A}$. These rule chains may not have been executed because one or more hypotheses in the rule chain were not satisfied. In the example rules stated above, assume that of the three rules given, Rule 1 is currently the only one in the knowledge base and that it did not fire because the expert system had no knowledge of the current temperature range attribute. In addition, assume that the correct diagnostic advice was to "replace part B6." If the missing attribute is not important to the performance of the rule, the error of omission can be corrected by generalizing the rule and removing the temperature range attribute. Depending on the complexity of the knowledge base and the rule chains generated, finding the rules to generalize and performing the generalization may be a complex process.

The knowledge base is incomplete when a rule cannot be found which is a candidate for generalization. In this case, the knowledge base does not contain information pertinent to the diagnostic action $d_i \in$ **C** – **A**. In this situation, a discovery learning mechanism should be invoked to capture the missing knowledge. Depending on the structure of the knowledge base, the discovery learning mechanism may be either simple or complex. One simple discovery algorithm uses the value of the input symptoms as the hypotheses of the new rule and the correct diagnostic action as the rule consequent [9].

The first step in improving existing rules is to identify which rules should be revised. This necessitates recording the trace of each rule chain producing system output along with corresponding rule unifications. Whenever the adaptive learner is invoked, one or both of the rule statistics must be updated. If the rule has participated in a rule chain leading to a component of set **C**, both of the rule's experience indicators should be incremented. If the rule has participated in a rule chain leading to a component of set **A** - **C**, only the times fired statistic should be incremented. The trace provides quick access to these statistics.

In cases where a rule chain terminates with an element $d_i \in$ **A** - **C**, the faulty rules must be identified. Bundy et al.[3] describe two basic techniques utilized by rule learning programs to identify faulty rules. Both approaches are similar in that they only identify the first faulty rule within a chain.

In the first approach, the actual rule chain is compared with the chain which should have fired. Some programs require this ideal chain as input [2] while others [6] attempt to derive it by analysis using problem-solving and inference techniques. The first difference between the chains indicates the faulty rule. The necessity of identifying the ideal rule chain makes this technique difficult to apply.

The second technique for finding a faulty rule is called Contradiction Backtracking. This technique, developed by Shapiro [8] does not require identification of an ideal chain. Assuming the actual rule chain concludes with d_i, Shapiro's algorithm begins by examining the last resolution step leading to d_i. If the propositions which were resolved to produce d_i are true, select the branch of the tree containing these propositions as part of the rule hypothesis, else select the other branch. Backtracking up the resolution tree continues in this manner until a rule from the rule base is reached. This is the faulty rule. Both Shapiro and Bundy et al. give examples of Contradiction Backtracking.

Unlearning strategies generally require an approach different from those described since deciding which rules to "unlearn" can not generally be determined by comparing the results of sets **A** and **C**. Unlearning strategies require that the experience indicators of each rule in the rule set be evaluated periodically. However, care must be taken not to remove a rule simply because of poor performance. While a rule that is correct six out of a thousand times is not contributing to the overall quality of the knowledge base, it may need to be modified and not merely discarded. In addition, rules having a small number of firings but a relatively high percentage of correct firings probably should be left in the knowledge base. An individual rule's high percentage of correct firings indicates it is participating in correct rule chains. An individual rule's low number of firings may indicate that the conditions it identifies are exceptional cases

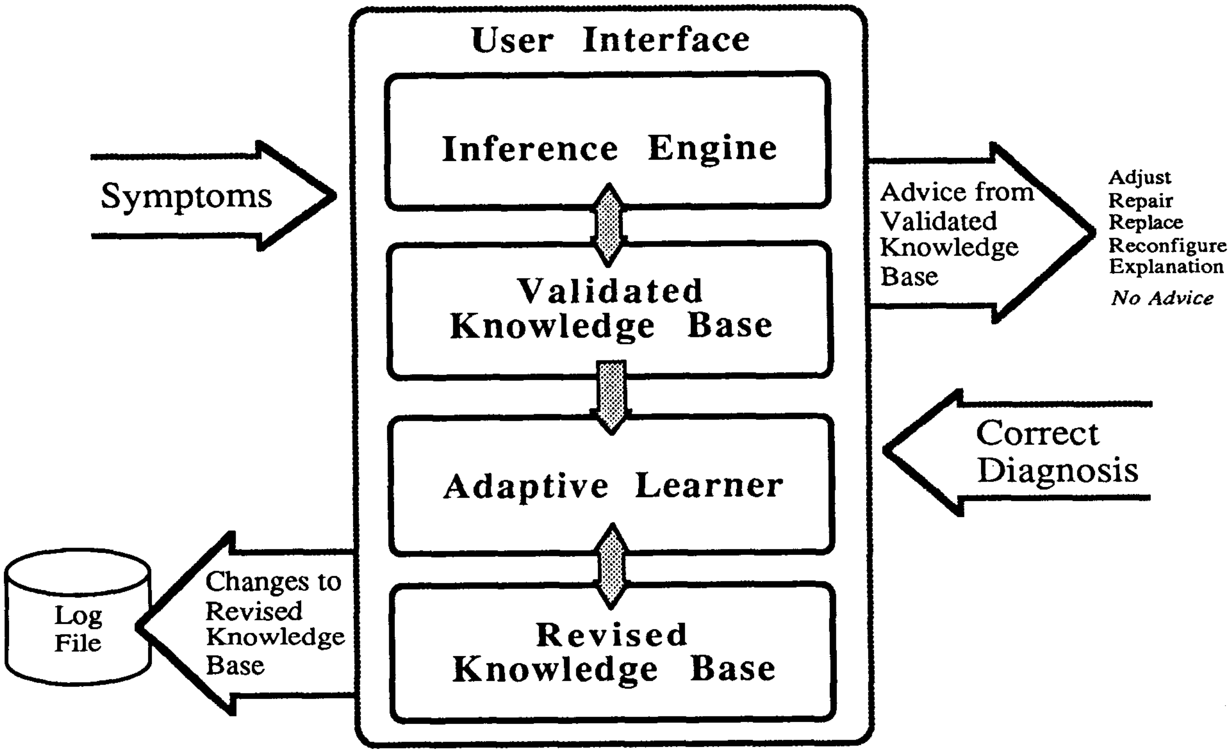

Figure 4. Dual Expert System Architecture.

and not merely noise in the input symptoms. If the times fired and times correct statistics are extremely small, the rule may have been created as the result of noisy input data.

The strategies described provide a means of incrementally updating the knowledge base each time a new piece of information is available. In diagnostic applications where human expert intervention is desired before changes are made to the knowledge base, it may be desirable to use a dual expert system as illustrated in Figure 4. The inference engine first reasons against the symptoms by using the validated knowledge base. It then outputs diagnostic advice. Next, the inference engine repeats the diagnosis using the revised knowledge base to produce a second set of advice. This advice need not be reported to the user. Upon receipt of the correct diagnosis, the adaptive learner updates the revised knowledge base. Knowledge base changes are logged so that they may be reviewed by domain experts. This scenario guarantees that the system always utilizes the validated knowledge base. However, at any point in time, human experts may review the revised knowledge base and, if desirable, use it to replace the validated knowledge base.

CONCLUSIONS

Techniques presented in this paper outline basic adaptive learning mechanisms and strategies for incorporating them into diagnostic expert systems. Although implementations of these strategies and learning mechanisms vary from system to system, the basic concepts are applicable to a large number of diagnostic expert systems. Continued research in these areas promises to reduce the maintenance costs of diagnostic expert systems.

REFERENCES

1. Blaxton T. A. and Kushner, B. G., *An Organizational Framework for Comparing Adaptive Artificial Intelligence Systems*, **1986 Proceedings of the Fall Joint Computer Conference,** IEEE Computer Society, November 1986, pp. 190-199.

2. Brazdil, P., *A Model for Error Detection and Correction*, Ph.D. Dissertation, University of Edinburgh, 1981.

3. Bundy, A., Silver, B., and Plummer, D., *An Analytical Comparison of Some Rule-Learning Programs*, **Artificial Intelligence**, Vol. 27, 1985, pp. 137-181.

4. Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D., *AutoClass: A Bayesian Classification System*, **Proceedings of the Fifth International Conference on Machine Learning**, Morgan Kaufmann Publishers, Inc., June 1988, pp. 54-64.

5. Markovitch, S. and Scott, P. D., *The Role of Forgetting in Learning*, **Proceedings of the Fifth International Conference on Machine Learning**, Morgan Kaufmann Publishers, Inc., June 1988, pp. 459-465.

6. Mitchell, T.M., Utgoff, P.E., and Banerji, R., *Learning by Experimentation: Acquiring and Modifying Problem-Solving Heuristics*, **Machine Learning**, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Ed.s), Palo Alto, CA: Tioga, 1983, pp. 163-190.

7. Quinlan, J.R., *Induction of Decision Trees*, **Machine Learning**, Vol. 1, 1986, pp.81-106.

8. Shapiro, E., *An Algorithm That Infers Theories From Facts*, **Proceedings of the Seventh International Joint Conference on Artificial Intelligence**, Los Altos, CA: William Kaufmann, Inc., 1981, pp. 446-451.

9. St. Clair, D. C., Bond, W. E., Flachsbart, B. B., and Vigland, A. R., *An Architecture for Adaptive Learning in Rule-Based Diagnostic Expert Systems*, **1987 Proceedings of the Fall Joint Computer Conference**, IEEE Computer Society, October 1987, pp. 678-685.

10. St. Clair, D. C., Bond, W. E., and Flachsbart, B. B., *Using Output to Evaluate and Refine Rules in Rule-Based Expert Systems*, **Proceedings of the Third Conference on Artificial Intelligence for Space Applications, Part I,**" NASA Conference Publication 2492, November 1987, pp. 9-14.

11. Utgoff, P. E., *ID5: An Incremental ID3*, **Proceedings of the Fifth International Conference on Machine Learning**, Morgan Kaufmann Publishers, Inc., June 1988, pp. 107-120.